Docket No. 22339-40020 U.S. Patent Application No. 09/944,131

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of Robert TISCHER	Examiner: Jonathan D. SCHLAIFER
Confirmation no. 2130 Serial No.: 09/944,131	Docket No.: 22339-40020
Filed: August 31, 2001	Group Art Unit: 2178
For: METHOD AND SYSTEM FOR PRODUCING AN ORDERED COMPILATION OF INFORMATION WITH MORE THAN ONE AUTHOR CONTRIBUTING INFORMATION CONTEMPORANEOUSLY	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. § 1.132

Sir:

1. I, Robert Tischer, am the inventor on the above-identified patent application.

2. I hold a Bachelor of Science degree in Electrical Engineering, a Master of Science degree in Computer Science and a Master of Arts degree in Psycholinguistics. In addition, I have been a C++ software developer for 12 years and a software developer for over 20 years. I am an expert in the fields of Learning Theory, Linguistics and Computer Science and was an expert at the time of the invention.

3. I am familiar with the prosecution history of the above-identified application and understand that U.S. Patent No. 6,061,697 (Nakao) and U.S. Patent No. 5,805,897 (Glowny) have been cited by the Examiner assigned to this application as rendering obvious the pending claims. I have read and am familiar with the contents of both the Nakao and Glowny references.

4. It is my understanding that the Examiner has taken the position that "it is impossible to verify that two events occur at exactly the same time" and therefore has interpreted the claim term "contemporaneous" to mean "happening within a reasonable time interval of one

another." This statement both acknowledges an important difference between the claimed invention and Nakao and misconstrues an important element of the claimed invention. With regard to the difference between the claimed invention and Nakao – for a particular Nakao system client to view the content edits by another author, a request is required (either explicitly or at a timed "interval") to update all of the document type declaration (DTD) element's content. This is a consequence of the Nakao system using the client-server model for distributed document control. *See, e.g.*, Nakao, Figure 3. It is well known in the art that client-server clients require refreshing (*i.e.*, are not synchronously and/or contemporaneously updated) when changes occur at the server. Therefore, Nakao clients lack the ability to contemporaneously view the document, except as a secondary process. In Nakao there is always a delay between the edit event and propagation to all of the DTD element's content. In contrast, the claimed system utilizes the cooperative functioning of the multiple fragment executables as one implemented DTD to push edit events, either actually or virtually, as they occur, without a "request" or specific "time interval."

In addition, Nakao's system contains no mechanism for DTD context change, except by way of conventional versioning. Versioning, by definition, is non-cooperative since all potential context contributions are suppressed between versions and furthermore, authors must appeal to the versioning entity in order to gain change access. As is the case with Nakao's content changes, these model variation changes can only be observed at the other author's sites upon request (preset or manual), due to the nature of the client-server model. For the claimed invention, since the executables all function complementarily with regard to the other author's executables, each author may alter content or context at will to the extent of his executable's functionality, without conflicting with other author's contributions.

As such, context as well as content immediacy is not only possible, it is inherent within the claimed system as change is an event that drives synchronizing changes at other replicate sites. Thus, loosely defining "contemporaneous" as "happening within a reasonable time" conflicts with the meaning of this term imparted by the supporting description, and indeed its plain meaning. It is this immediacy (and the means utilized to produce it) that distinguishes the claimed invention from Nakao. In the claimed invention, any author in a group of authors using the invention has the ability, not only to make content and context changes to the group

document, but to always make them from the backdrop of all the other author's latest contextual as well as content contributions.

5. With regard to the Glowny reference, this reference does not teach the claimed peer-to-peer environment involving pushing context and content edit changes to document replicates. Glowny's system makes requests of remote operations and thus uses the remote-procedure-call (RPC) model, which is inherently client-server in construction. The strength of client-server is that its clients know what piece they want to retrieve from the server and consequently what to do with it. It is well known in the art that client-server client views are different from each other, and are, as such, not capable of being replicates. Glowny teaches that software can respond to a request that has been copied and installed on a remote system, which installed software is comprised of a particular operation. Glowny further teaches that execution of the now remote operation can then be accomplished using a remote command execution procedure (i.e., RPC). The remote execution of operations of Glowny contrasts with the claimed limitation of allowing multiple authors to make changes in a single distributed document contemporaneously. No author in the currently claimed system is remotely executing operations on another author's system because author changes are pushed as they occur.

Glowny does not teach any hierarchy of execution and its remote operations have no order associated with each other, which are requirements for placing the operation in a document context hierarchy constrained to a particular order, which, in turn, is a prerequisite for having a contemporaneously viewable document. In the claimed invention, installing and executing remote operations occurs within the context hierarchy after which, the resulting executables inherently function complementarily and cooperatively – but always locally – in tandem with the other DTD element executables.

6. If the *Nakao* system were converted to a peer-to-peer (P2P) system, as the Office suggests (the hypothetical "P2P Nakao system"), the result would have unacceptable performance if limited to content changes and would be inoperable if extended to DTD context changes. Since a P2P Nakao system would no longer have any mechanism for identifying which document piece an author is making changes to (in client-server systems, it is the client that indicates what document piece on the server he wants to check-out and subsequently make

changes to), each P2P Nakao system author would necessarily have a copy of the entire tagged document.

To illustrate certain differences between the claimed system and the P2P Nakao system proposed by the Examiner, the following hypothetical scenario is provided. Initially, all authors in a P2P Nakao system could show the complete and correct document since all P2P Nakao system authors would have an accurate replicate of the document. Thereafter, an event occurs where one author makes a content change to the document. A period of time goes by and a second author wants to make a content change. The second author does not know about the first author's change because his "replicate" document is out-of-date with respect to the first author's change. The first author then sends out his content change to the other authors in the form of a marked-up, i.e., tagged, document which contains his change. The recipients thereafter replace their version of the tagged document with the received (marked-up) document and all authors, once again, show an accurate replicate of the document with the first author's change included. In a more realistic scenario, assume now that a second author enters a change at the same time as a change is being made by the first author, and the second author sends his version of the tagged document out to replace the version viewed by the first author. Because of varying network delays and/or because the second author does not integrate the first author's changed document into his, the changed document propagated by the second author would prevail, wiping out the first author's change. Alternatively, the second author could choose to integrate the first author's changes with his partially completed changes – by either integrating his changes in the received first author's document or integrating the author's changes into his document. However, the first author's document has no mechanism for specifying where the changes were made, so the second author would necessarily have to adopt the first author's document and apply his changes to the newly arrived first author's document before proceeding with his changes in order to once again become a replicate. If he didn't do this, order of reception would determine which author's change – the first or second – would prevail and thus be viewed by the P2P Nakao system authors. Obliterating or suppressing other author's changes is not a cooperative way of maintaining replicate behavior. Without the Nakao server to impose its form of one-at-a-time check-out/check-in concurrency, the P2P Nakao system authors' replicates as they grow in size and number, would fail because they would copy over each other's changes, and have unacceptable performance due to (1) the inordinate amount of redundant data sent

across the network for each change, (2) the constant interruption in the editing process, and (3) the need to continually merge and resolve potentially conflicting changes.

Depending on whether both authors are simultaeously editing the document and whether additional authors are also editing the document, the scenario becomes exponentially more complicated and assures that each of the multiple authors cannot view edits made by other authors contemporaneously. Moreover, the above scenarios assume a static DTD. If a first P2P Nakao system author enters a change in the DTD context this first author's document structure no longer matches the other authors' document structures (i.e., is not a structural replicate). At best, each P2P Nakao system recipient would have to correlate the tagged document with the received document in order to match up the elements and detect the difference in structure.

The claimed invention solves these content sharing and structural change problems by dividing up the elements of the system before any author contributes so that no two authors own the same element in the document tree at the same time. Thus, unambiguous change and contemporaneous viewing is provided in the claimed system and is absent in the P2P Nakao system proposed by the Examiner.

7. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Respectfully submitted

Date: August 10, 2005

Robert Tischer

Robert Tischer